**6 FEATURES TO BE TESTED**

A. Graphical interface in check sheet form, for faculty advisor to view, edit and update advisee’s academic information.

B. Recommend course schedule for upcoming semester.

C. Provide alerts if advisee is at risk of deviating from advisee’s planned graduation date.

D. Keep track of advisee’s academic progress and store student academic records.

E. Generate information to help advisor’s fill out advisee forms.

F. Allow advisor to store notes about student.

G. Retrieve and update course information, semester offerings, university requirements, and department check sheets.

**7 FEATURES NOT TO BE TESTED**

A. Network security and internet access.

B. Minors are not integrated into the system, and therefore will not be tested

**8 APPROACH**

**8.1 Testing Levels**

The testing for the Student Advisement System will consist of Unit and Integration test levels. There are three modules and three developers. Each developer will be assigned a module to test that he/she did not implement themselves.

UNIT Testing will be performed by a tester who did not write the module and tests for that module. Unit test cases must be well documented (test case list, sample output, data printouts, defect information) before being accepted and passed on to the test person. All unit test information will also be provided to the test person. The test person will record their results and send them back to be analyzed and evaluated by the developer of the module.

INTEGRATION Testing will be performed by all three of the project developers. Modules should have no major defects before being accepted for integration testing.

**8.2 Configuration Management/Change Control**

All system files reside in the project’s Github repository [https://github.com/kusoftwarefun/Student-Advisement-System]. Github has an extensive configuration, version, and change management system. All changes, enhancements and other modification requests to the system will be handled through the Github interface, and typed out in a standard format; purpose of issue, high level description of the issue, low level description of the issue, priority level of issue (minor, major, critical ), unique ID for issue.

**8.3 Test Tools**

* CSIT Course Dependencies Graph
* CSIT Checksheets
* Browser

**8.4 Meetings**

The team will meet once every two weeks to evaluate progress to date and to identify

error trends and problems as early as possible. Additional meetings can be scheduled as required, and online communication is always available.

**8.5 Measures and Metrics**

The following information will be collected by the Development team during the testing

process. This information will be provided to the test team at program turnover as well as be

provided to the project team on a weekly basis.

1. Defects by module and severity.

2. Defect Origin (Requirement, Design, Code)

3. Time spent on defect resolution by defect, for Critical & Major only. All Minor defects

can be totaled together.

**9 ITEM PASS/FAIL CRITERIA**

At any phase testing is complete when the only remaining defects in the system are categorized as minor.

**10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS**

Generally not applicable. The team will use common sense, such as not testing data sets that are virtually equivalent,. As such, tests should always be yielding valuable information, pass or fail.